

WHAT IS CLAIMED IS:

- Sub B1
1. A computer-implemented method, comprising:
disabling interrupts on a computer system;
preparing a first set of data for writing;
5 instructing a controller to write data asynchronously to a
disk;
polling, in intermittent polling operations, a status
register to determine when the write to the disk is complete;
and
10 while between polling operations, preparing a second set
of data for writing.
2. The method of claim 1 wherein preparing the first set
of data for writing includes compressing the data.
- 15 3. The method of claim 1 wherein preparing the first set
of data for writing includes segmenting the data.
4. The method of claim 1 further comprising, calling at
20 least one driver to obtain information about write limitations
of the disk.
5. The method of claim 1 further comprising, reserving
at least one special section of memory for preparing the first
25 set of data for writing.

Cont
B1

6. The method of claim 1 further comprising, calling a disk driver to obtain information about memory requirements of the driver.

5

7. The method of claim 6 further comprising, reserving at least one special section of memory for the disk driver based on the memory requirements thereof.

10

8. A computer-implemented method, comprising:
disabling interrupts on a computer system;
instructing a controller to read data asynchronously from a disk to an input buffer;
polling, in intermittent polling operations, a status register to determine when the read from the disk is complete;
and
while between polling operations, decompressing at least some data in the input buffer.

20

9. The method of claim 8 further comprising, loading at least one driver to enable the controller to read the data asynchronously from the disk.

25

10. The method of claim 8 wherein decompressing at least some of the data in the input buffer comprises, accessing the

data including tokens comprising literal symbols and length,
offset pairings, ensuring that space to hold a predetermined
number of at least two output symbols is present in an output
buffer, selecting a token from the input data, and when the
5 token comprises a literal symbol, copying the literal symbol
into the output buffer without checking whether the symbol will
fit into the output buffer, or when the token comprises a
length, offset pairing, copying at least the predetermined
number of symbols into the output buffer without checking
10 whether each symbol will fit into the output buffer.

11. A system for entering a computing device into
hibernation, comprising,

a non-volatile storage device;

15 a non-volatile storage device controller configured to
write data to the non-volatile storage device;

a DMA controller connected to the non-volatile storage
device controller, the DMA controller configured to access
memory of the computing device and direct the non-volatile
20 storage device controller to write data at the accessed memory
to the non-volatile storage device;

a compression engine configured to compress data from one
memory location to an output buffer; and

a hibernation engine configured to direct the compression engine to compress an amount of uncompressed data to the output buffer, to communicate with the DMA controller to request a write of compressed data from the output buffer when compressed data in the output buffer is ready for writing, to request a status of the write operation that is present in a status register, and to communicate with the compression engine to compress another amount of uncompressed data to the output buffer while the status indicates that the requested write is pending.

12. A computer-implemented method, comprising,
accessing compressed input data including tokens
comprising literal symbols and length, offset pairings;
ensuring that space to hold a predetermined number of at
least two output symbols is present in an output buffer;
selecting a token from the input data; and
if the token comprises a literal symbol, copying the
literal symbol into the output buffer without checking whether
the symbol will fit into the output buffer; or
if the token comprises a length, offset pairing, copying
at least the predetermined number of symbols into the output
buffer without checking whether each symbol will fit into the
output buffer.

B1
13. The method of claim 12 wherein ensuring that space to hold a predetermined number of at least two output symbols is present includes, dividing space remaining in the output buffer by the predetermined number.

5

14. The method of claim 12 wherein the token comprises a length, offset pairing and wherein copying a number of symbols comprises, copying the predetermined number regardless of an actual length value in the length, offset pairing, and
10 adjusting a pointer based on the actual value.

15. The method of claim 12 further comprising,
determining that space to hold a predetermined number of at least two output symbols may not be present in an output
15 buffer, and invoking another decoder to decompress the input data into the output buffer.

16. A computer-implemented method, comprising,
receiving a first request to preserve an existing state of
20 a computer system;

maintaining sufficient power to a volatile memory of the computer system to preserve memory contents therein; and

writing information corresponding to at least some of the memory contents to a non-volatile storage of the computer

system, the information sufficient to restore the state of the computer system from the non-volatile storage.

17. The method of claim 16 further comprising, receiving
5 a second request to restore the computer system to a state corresponding to the existing state when preserved, and if power was interrupted to the volatile memory, restoring the state of the computer system from the non-volatile storage.

10 18. The method of claim 17 wherein writing information corresponding to at least some of the memory contents includes compressing data, and wherein restoring the state of the computer system from the non-volatile storage includes decompressing the data.

15 19. The method of claim 16 wherein writing information corresponding to at least some of the memory contents includes, disabling interrupts on a computer system, preparing a first set of data for writing, instructing a controller to write data
20 asynchronously to a disk, polling, in intermittent polling operations, a status register to determine when the write to the disk is complete, and while between polling operations, preparing a second set of data for writing.

20. The method of claim 19 wherein preparing the first set of data for writing includes compressing the data.

21. The method of claim 20 further comprising, restoring
5 the state of the computer system from the non-volatile storage,
including decompressing the data.

Add
B1

10

PREFACE